

Composites and Mashups and Processes, Oh My!



Lori MacVittie, 2007-14-10

By **Lori MacVittie**

In the increasingly buzzword compliant world of SOA and Web 2.0 it is inevitable that you have, or will, come across three of the industries' favorite terms: composites, mashups, and processes. These terms represent concepts that while similar actually describe very different uses of services.

The distinctions between the three are subtle and in many cases it makes almost no sense to differentiate between them. But one of the goals of SOA is to provide a common language through which IT and business folks can communicate. In order to maintain that framework it is necessary to distinguish between the differing recombination techniques that form the basis for building out useful, services-based applications.

Composites

Composites should more accurately be referred to as "composite applications". Composite applications are SOA-based applications built upon multiple services where the recombination of data from multiple services occurs within the application. It is usually the case that the data being combined is related in some way to a common business entity. For example, customer data retrieved from a CIS (Customer Information System) is combined with customer-specific orders retrieved from an order entry system. The data combined are related to the customer, and is used to present a holistic view of the business entity, in this case the customer.

Composites are based upon traditional integration applications in that they retrieve information from and execute business logic in other applications, or in the case of SOA and composites, other services. The recombination of data, i.e. the joining of data and information from multiple services, happens inside the application. In the case of web-based applications this is usually on the server and in the case where fat clients are involved some of that recombination or logic processing may occur on the client. The primary factor here is that the application controls the flow of data and logic, making it a different beast from a process.

Processes

Processes uses services in much the same manner as that of a composite, but does so using an intermediate execution language such as BPEL (Business Process Execution Language) that is usually deployed on its own server. Processes are not as robust as composites in that their ability to execute business logic is limited to what can be described by a language such as BPEL or reasonably automated using the process management solution's environment.

Processes tie together disparate business services to execute a business process with minimal, if any, manual intervention. This is often referred to as orchestration.

Mashups

Mashups tend to be more oriented on the presentation layer of an application. The most common example of a mashup is coming mapping data – often from Google Maps – with some other geographical-based data in order to better visually represent the data. Mashups tend to combine disparate data and often perform the recombination on the client rather than on the server, which is different from the related-data recombination that occurs in a composite application.

All three of these types of applications share one thing in common: they rely on *services* to provide data. These services are always external to the application and therefore require network connectivity in order to be of use. This means that all three types of applications not only incur the performance penalties associated with the increased overhead of managing these network connections, but also that any one of the services used to build the application becomes a single point of failure.

The reliance on external services makes applications increasingly complex, as transactions may occur – or not occur – externally and the application logic must be prepared to respond and, one hopes, recover from such events. Reliability of the services upon which these applications are built is paramount, and regardless of what name you use to refer to such an application the same underlying issues must be addressed either by the developer, by the network architect, or both.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com

©2016 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113